CEDIO_B

5-apr-2011.
Embedded software version 2.

1. Main features

The device is based on unified digital input/output board (CEDIO) and customized by embedded software. CEDIO_B is intended for using as multifunctional device in HVT of coolers.

The device includes:

- 16-bits output register with power outputs;
- programmable generator connected with pin OUT7;
- 16-bits input TTL register;
- CANBUS interface for interaction with control computer;
- On-board micro-controller.

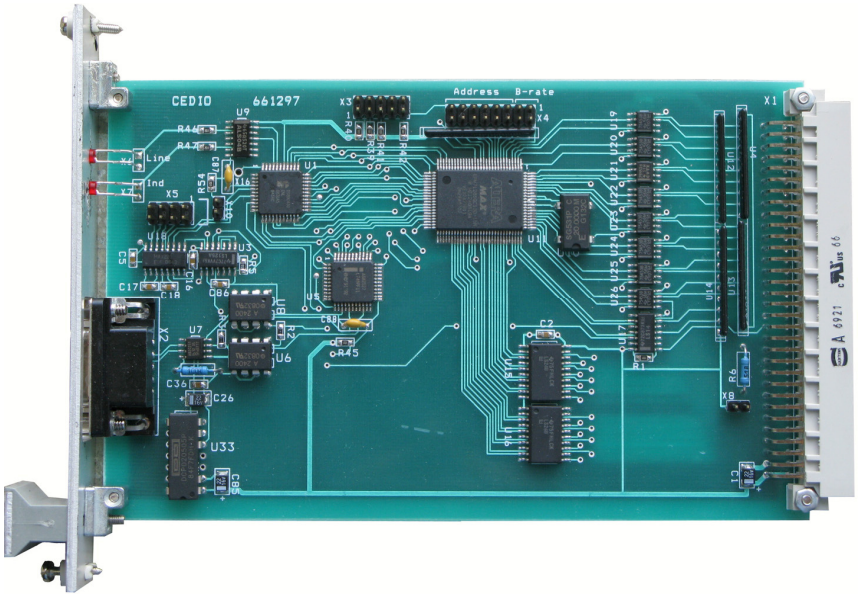


Photo of CEDIO.

The device is intended to be embedded in power supply racks. The device requires for proper operation the only power supply with voltage +5V (±5%).

2. Specifications (main parameters):

1. Channels of output register – 16.
2. Maximal voltage for output register - 30 V.
3. Maximal current for output register - 300 mA.
4. Channels of input register – 16.
5. Voltage for input register- TTL levels or contact with “ground”.
6. Pins for phase output (routine 0) – 2.
7. Duration of phase pulses (routine 0) – 1 mS – 65 Sec.
8. Pulse channel – 1 (OUT7).
9. Duration of pulses on OUT7 (routines 0,1) – 200 nS – 6 Sec.
10. CANbus transceiver is galvanically isolated from network and it is in compliance with ISO 11898-24V (chip PCA82C251).
11. Voltage between transmission line and device- 1000V.
12. Baud rates- 1000, 500, 250, 125 Kbaud (may be chosen by jumpers).
13. Voltage of power supply- +5V, ±5%.
14. Power supply current- <0,9 A (typical value- 0.6A). Power supply must provide starting current > 1,0 A.
15. Size of device - 3U * 160 mm.

3. External connections

The device CEDIO is implemented as eurocard 3U*160. A front panel of the device contains a network connector (DB-9M) and two LEDs. One LED is blinking during processing CANbus messages. The second LED using depend on embedded software version. Connection with external signals carry out by rear connector.

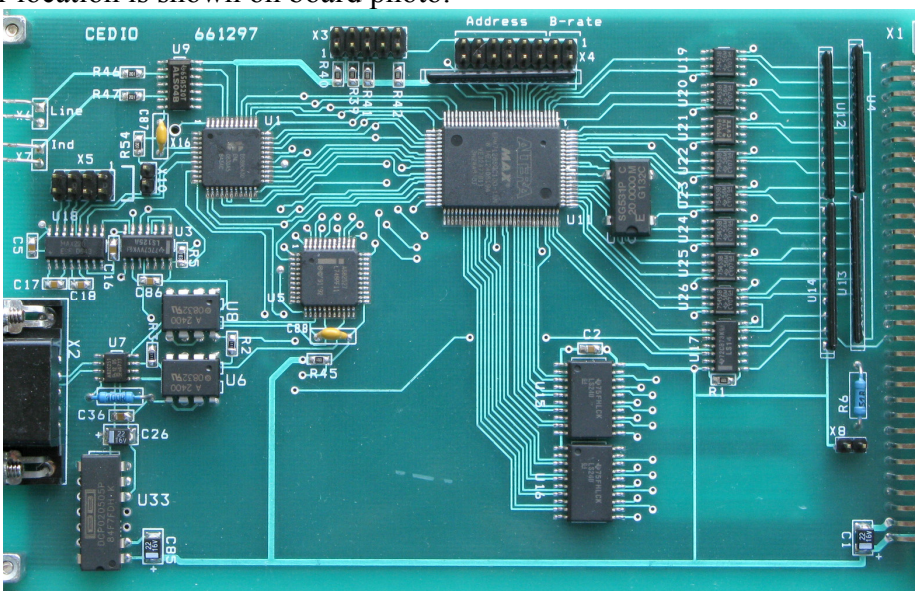
3.1. Jumpers

The device CEDIO has jumper X8 and set of jumpers- X4.

X8 is intended for using internal power supply (+5 V) instead of external for powering load of output register (see description of output register).

X4 includes 8 jumpers. 6 jumpers define number (address) of device in network (this number is used to compose identifier of messages) and 2 jumpers define baud rate.

Jumper location is shown on board photo.



Destination of jumpers in X4 group.

Designation	Location	Destination
X4-7	Upper	N5- included in device number (most significant bit)
X4-6	…	N4- included in device number
X4-5	…	N3- included in device number
X4-4	…	N2- included in device number
X4-3	…	N1- included in device number
X4-2	…	N0- included in device number (least significant bit)
X4-1	…	BR1- defines baud rate
X4-0	Lower	BR0- defines baud rate

Jumpers N5…N0 defines logical number (address) of device which is used to compose message identifier for CANbus network (for more detail see PROTOCOL part of this description). An installed jumper should be interpreted as logical 0 and absence of jumper should be interpreted as logical 1.

Don’t use addresses 34, 3C, 3D, 3E и 3F (hexadecimal).

Baud rate coding.

BR1	BR0	Baud rate
Connected	Connected	1000 Kbit/sec
Connected	Disconnected	500 Kbit/sec
Disconnected	Connected	250 Kbit/sec
Disconnected	Disconnected	125 Kbit/sec

NOTES:

1. CANbus is bus with multiple accesses and incorrect baud rate setting may affects on transfer messages of other devices up to impossibility of access to the device.
2. In network may exist concurrently devices with identical numbers (addresses). Formally it is permissibly, but actually it does cause a lot of problem. Connecting to network devices with identical numbers is strictly not recommended.

3.2 Front panel

A front panel includes:

Line LED

Ind LED

CANbus connector

Line LED is blinking during processing CANbus messages by onboard processor.

Ind LED is ON when device executes routine 0 or 1.

After power-on the device blinks by all LEDs a few times. Then the processor indicates version of device. Version CEDIO_B is marked by double flash of **Ind** LED. Both LEDs are blinking during "bus-off recovery" procedure also.

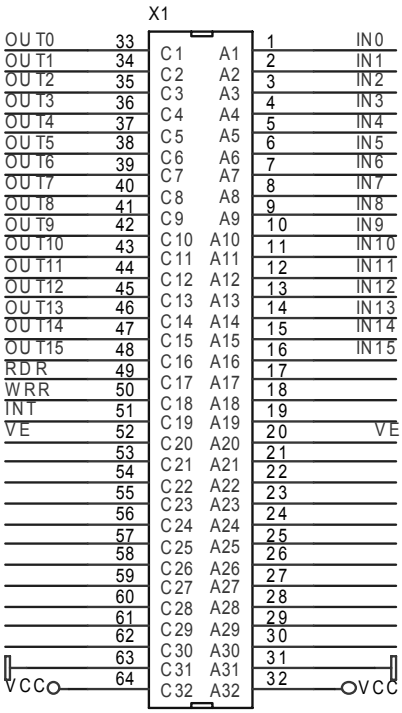
CANbus connector (DB-9M) is intended for connection to media. Pin designations follow below in table.

2	CAN-L	One wire in pair
3	GND	Shield of cable
7	CAN-H	One wire in pair

Shielded twisted pair is used as media. According to the ISO 11898-2 it should has nominal impedance 120 Ohm. Line termination has to be provided by termination resistors of 120 Ohm located at both ends of the line.

3.3 Main connector

There is DIN41612 connector for connecting with input and output signals.



The main connector provides for user register inputs, outputs and pins for powering device. The device uses the only external power supply +5B (5%). For powering device one should use pins 31, 32, 63 and 64.

The following outputs have special destination.

Pin	Destination
OUT0	Phase 0 of routine 0
OUT1	Phase 1 of routine 0
OUT7	Blank signal / Output of pulse generator

Output register is implemented on power logical elements with open collector and can control by electromechanical relay. Keep in mind that output register has not protection circuitry and if load has inductance (relay, for example) user must add diode to remove spike. Register outputs with VE (external voltage) by pull-up resistor 10 KOhm. If register will be used as TTL register then pin VE can be connected with internal power supply (+5 V) by X8 jumper.

Zero in output register corresponds to disconnected output (no output current).

Input register is intended for TTL levels. Each input is connected with power supply by pull-up resistor (10 KOhm). It may be used for reading mechanical contact state, accordingly. Disconnected input is interpreted by device as logical zero.

4. Basics of operations for CEDIO_B

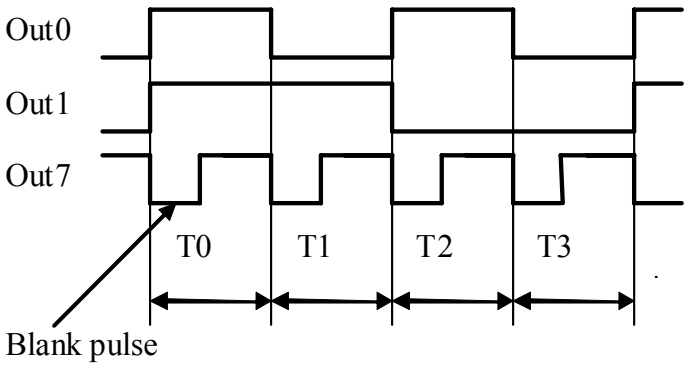
The device has three states:

- Passive
- Routine 0 execution
- Routine 1 execution

In passive state the device is digital input/output register with power outputs (bits OUT0,1,7 are not used). On receiving Start command (F7) with code the goes in “routine execution” state until receiving Break command (FB). During routine execution all general purpose bits of output register are available for user. Input register can be used as ordinary input register with reading input pins and transfer this information to computer.

Routine 0.

On receiving Start command the device goes into sequence phase generation state. Duration each phase should be written before. On phase changing the device generates pulse on OUT7 with preprogrammed duration. The routine execution will be completed on receiving Break command. All output signals are inverted (negative logic).



Duration each phase defined independently from other phases (separate registers). To exclude any phase user should write zero in corresponding register. Duration of the Blank pulse is the same for any transition.

Real sequence of phase values is 0 – 1 – 0 – 2 and so on.

There is possible to define duration of Blank pulse more than duration of phase. The device interprets it some way.

During routine 0 execution bits OUT0 and OUT1 have phase code value, OUT7 bring out Blank pulse and OUT2-6 are cleared. During this routine microcontroller ignores write command to low byte of output register.

Routine 1.

During routine 1 execution the device connect to OUT7 pulse generator. Period of pulses defined by “phase 0 duration” register. Duration of this pulse is defined by “generator” register. The routine execution will be completed on receiving Break command.

During routine 1 execution writing lo low port of output register is possible, but bits 0,1 and 7 will be cleared by processor.

5. Protocol (interaction with CANbus) of CEDIO_ B

Identifier bit definition.

Identifier bits	ID10…ID08	ID07...ID02	ID01…ID00
Bit field	Field 1	Field 2	Field 3
Destination	Priority	Address	Reserve

Comments to bit distribution:

Field 1 – priority field (type field):

Code 5 – a broadcast message (field 2 is ignored).
Code 6 – ordinary (address) message.
Code 7 – response (reply for type 6 message).

Code 0 is forbidden, other combination is not used (they are reserved for future extensions).

Field 2 – a physical address field. It defined address device (this address is defined by jumpers on-board). Don't use addresses 34, 3C, 3D, 3E and 3F (hexadecimal).

Field 3: User should set zero in this field. The device can send messages with different values in this field.

Any device on receiving address message interprets information by its content. If received message requires a reply, the device sends required information by message with code 6 (response type message). Broadcast messages should be received by all devices simultaneously and required actions should be done in all devices. The device doesn't check size of message except for F4 command (sequential write to file).

Size of transmitted messages is not defined and may be different in different versions of device.

Data interpretation.

Receiving CANbus message device interprets data the following way: the first byte (byte 0) is interpreted as descriptor of message (command) and the following bytes are parameters.

80÷83 – write to "phase duration" register (phases 0÷3)
84 – write to generator register
E8 - request for data from input and output registers
E9 - write to output register
F7 – start routine execution
FB – break routine execution
FE - device status request
FF - device attributes request

5.2. Detail description of messages (commands) (all codes are hexadecimal)

Message 80 - 83 – write to "phase duration" register (phases 0÷3), then 2 data bytes.

Examples:

83	12	1
3rd phase	Byte 0	Byte 1

This message defines 3rd phase duration as 274 mSec (1*256+12).
Byte 1 is high byte and byte 0 is low byte.

Message 84 – write to pulse generator register, then 2 data bytes.

84	Quantum	Time

Quantum – time quantum code, see table below.
Time – code of pulse duration. Real time is calculated my multiplication this code and time quantum used.

Code	Quantum weight
0	200 nS
1	400 nS
2	800 nS
3	1.6 uS
4	3.2 uS
5	6.4 uS
6	12.8 uS
7	25.6 uS

Message E8 - request date from registers. This message has not additional information. In reply a device sends a message with output register bytes and input register bytes.

E8	DO0	DO1	DI0	DI1	0	0

Here:
DO0 – data from output register (bits OUT0-OUT7);
DO1 – data from output register (bits OUT8-OUT15);
DI0 – data from input register (low byte, bits IN0-IN7);
DI1 – data from input register (high byte, bits IN8-IN15);

Message E9 - write data to output register.
Bytes 1 and 2 contain information to be written into output register (bits OUT0-OUT7 and OUT8-OUT15, respectively).

Note:

Writing to high port is available always but writing to low port bill be performed depend on device state (see “bacics of operation”).

Message F7 – start of routine execution
Byte 1 is parameter (number of routine).

Message FB – break of routine execution

FE – запрос статуса

На эту команду в ответ выдается следующее сообщение.

FE	Status	Valid

Status contains a set of flags.
b0,b1 – correspond to current phase and states of OUT0 and OUT1, respectively.
b2 – status of process (1- routine executed, 0- passive state).
b4-b7 routine number (0, 1).

Message FF - device attribute request. There are not additional parameters. In reply a device sends the following message:

FF	Device Code	HW version	SW version	Reason

Device Code- device type (for CEDIO_B it is equal 29).
HW version- hardware version of device.
SW version- software version of device.
Reason- reason of sending this message:

0 - After power-up.
1 - After reset by button on front panel.
2 - On request by address message with code FF.
3 - On request by broadcast message (who is here?).
4 - On restart by Watchdog timer.
5 - On busoff recovery.

5.3. GLOBAL messages (broadcasts)

For broadcast messages all devices analyze only field 1 in CANbus identifier. Valid combination is 5. A first byte of date presents a broadcast command. CEDIO_B uses the following broadcast commands:

FF- request “Who is here”. On this broadcast request all devices on-line must send into network message with their attributes (and identifier).

7. Software versions for CEDIO_B

Version 2.

1. Routine 1 is added. In version 1 was forbidden writing to low port of output register. Now it depends on device state.